Maria K. Nelson (State Bar No. 155608)
JONES, DAY, REAVIS & POGUE
555 West Fifth Street
Suite 4600
Los Angeles, California 90013-1025
Telephone: (213) 489-3939
Facsimile: (213) 243-2539

Blaney Harper (*Pro Hac Vice*)
Laura Talley Geyer (*Pro Hac Vice*)
JONES, DAY, REAVIS & POGUE
51 Louisiana Avenue, N.W.
Washington, DC 20001-2113
Telephone: (202) 879-3939
Facsimile: (202) 626-1700

Attorneys for Plaintiff
NETWORK CACHING TECHNOLOGY, L.L.C.

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

NETWORK CACHING TECHNOLOGY, L.L.C., Plaintiff, v. NOVELL, INC., VOLERA, INC., AKAMAI TECHNOLOGIES, INC., CACHEFLOW, INC., AND INKTOMI CORPORATION, Defendants.	Case No. CV-01-2079 (VRW)

DECLARATION OF JOHN C. MITCHELL, PH.D.

1. My name is John Clifford Mitchell. I have been a practicing computer scientist for over twenty years. Since the beginning of 1988, I have been an Assistant, Associate, and (full) Professor of Computer Science at Stanford University. My general research areas involve

the study of programming languages, software development methods, type systems, object systems, formal methods, and computer security. These areas of research involve the relationship between program structure and the functions performed by a program. A more detailed review of my qualifications in the field of computer science is set forth in my Curriculum Vitae, attached as Exhibit A.

2. I have been engaged by Networking Caching Technology, L.L.C. ("NCT") to analyze and provide my opinion on certain general computer programming issues. In particular, I have been asked to explain whether it is generally possible to determine the location within software of the instructions for carrying out a function, given only information about the externally observable behavior of a system carrying out this function. That is, I have been asked to assume that software exists which causes a computer (or "black box") to perform certain observable functions. I have then been asked whether a specific programming structure (*i.e.*, specific instructions within the software) may be determined based on the observable behavior of the black box.

3. To answer the above question, I believe a review of certain general computer software concepts is useful. Computer software is produced by a series of steps. Initially, a designer formulates a set of requirements. For example, the requirements for a word processor may specify that the word processor displays a text window on the user's screen, the user may type text in the window, and the user may print the contents of the window by clicking on a print icon. After the requirements are specified, software engineers decide on a set of program modules and their interfaces. One word processor module might manage printing while another contains the algorithm for deciding where to break lines on a page. Many software systems use library modules that were written for general use by many programs. After the modules and their

interfaces are determined, programmers write the code that causes each module to perform as designed. The code is then translated from a human readable programming language, such as the C or C++ programming language, into machine language (generally not human readable) by a compiler. Modern compilers translate a single C or C++ instruction into many machine language instructions. In addition, a compiler might reorder statements to make the software run faster, or copy a section of the code from one module into another (a transformation called in-lining) to avoid the cost of a function call when the software is executed.

4. Based on my experience and research, it is my opinion that the determination of a specific location within software that causes a black box to carry out a particular observable function may not be uniquely determined based on the observations of the function. The reason for this is that there are many equivalent ways to program any one particular function. While observing the black box as it operates (so called "reverse engineering" the software) may indicate that the program has a number of identifiable functions and characteristics (*e.g.*, receiving certain types of data, processing data, messaging, etc.), these observations cannot uniquely identify the location within the program of a series of instructions for carrying out any particular function. That is, from observations, it may be clear what happens and why, but the specific order of instructions will not be determinable.

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

Executed this 5th day of June, 2002, in Palo Alto, California.

John C. Mitchell

Curriculum Vitae
John Clifford Mitchell
April 11, 2002

General Research Interests

Computer security: access control mechanisms, cryptographic protocols and mobile code security. Programming languages and software development methods, type systems, object systems and formal methods. Applications of logic to computer science.

Personal

Born December 20, 1955 in Palo Alto, CA.
U.S. citizen, married, two children.

Address

Home 845 Esplanada Way
Stanford, CA 94305

Office Department of Computer Science
Stanford University
Stanford, CA 94305-9045
(415) 723-8634

Net mitchell@cs.stanford.edu http://www.stanford.edu/~jcm/

Education

Ph.D. Computer Science, MIT, August, 1984. Thesis title: *Lambda Calculus Models of Typed Programming Languages.* Supervisor: A.R. Meyer.

S.M. Computer Science, MIT, January, 1982. Thesis title: *Axiomatic Definability and Completeness for Recursive Programs.* Supervisor: A.R. Meyer. Entered MIT September, 1980.

B.S. Mathematics with Distinction, Stanford University, June, 1978. Transferred September, 1976 from University of Wisconsin, Madison.

Research and Teaching Positions

September, 1997 – present:
Professor, Department of Computer Science, Stanford University.

September, 1990 – 1997:
Associate Professor, Department of Computer Science, Stanford University.

July–December, 1995:
Visiting Scientist and Program Co-organizer, Newton Institute for Mathematical Sciences, Cambridge University.

January, 1988 to September, 1990:
Assistant Professor, Department of Computer Science, Stanford University.

September, 1984 to January, 1988 and Summer, 1983:
Member Technical Staff, Computing Science Research Center, AT&T Bell Laboratories.

Spring 1986:
Adjunct Assistant Professor, Dept. of Computer Science, New York University.

June, 1978 to August, 1980 and Summers 1976, 1977:
Research Engineer, University of Wisconsin Solar Energy Laboratory.

EXAMPLE CONSULTING POSITIONS

Jupiter Media Metrix/Brobeck Phleger & Harrison: Technical expert for patent suit, Feb 2002 – .

Biometric Access Corporation/Brobeck Phleger & Harrison: Technical expert for patent suit, Jan – Feb 2002. Declaration and deposition.

Third Voice/Heller Ehrman White & McAuliffe: Technical expert for patent dispute, Sept – Oct 2000.

Lucidity/Mitchell Silberberg & Knupp: Technical expert for contract dispute, June – Sept 2000.

Xerox PARC: Research on trust management, June – Oct 2000.

Trend Micro/Townsend and Townsend and Crew: Technical expert for patent infringement suit, including two expert reports, deposition, and testimony at jury trial, Sept 1999 – May 2000.

Kestrel Institute: Research on principles of computer security, logical methods for protocol analysis, Feb 1999 – present.

Airtouch Corporation: Mathematical analysis of accounting methods related to international tax law, Dec 1997 – Aug 1998.

Neuron Data/Brobeck, Phleger & Harrison: Software copyright and contract dispute, October 1996 – January 1997.

Pure Software/Brobeck, Phleger & Harrison: Software patent cases, copyrights, contract disputes; July, 1994, January–March 1995 and June–August 1996.

Interval Research: Scripting languages for internet applications, May-June 1995.

Kestrel Institute: Specification language and type/module system for prototyping language. February – May, 1992.

Hewlett-Packard Labs: ABEL project to develop typed, object-oriented programming language and methodology. May, 1988 – May, 1990.

Research Contracts, Grants and Awards

NSF Information Technology Research (ITR), 2001-2006. Computational Logic Tools for Research and Education. David Dill (Stanford, PI), Zohar Manna (Stanford), John Mitchell (Stanford).

ONR URI Program, 2001-2004. Software Quality and Infrastructure Protection for Diffuse Computing, ONR Grant N00014-01-1-0795. Joan Feigenbaum (Yale), Joseph Y. Halpern (Cornell), Patrick D. Lincoln (SRI), John C. Mitchell (Stanford), Andre Scedrov (U Penn, PI), Jonathan M. Smith (U Penn).

DARPA99-33-034, 2000-03. Agile Management of Dynamic Collaboration. J. Mitchell, Principal Investigator; P.Lincoln (SRI), Co-PI; M. Baker (Stanford), D.Dill (Stanford), L. Gong (JavaSoft).

DERA, 1999-2000. Instrumentation and Checking Techniques Applied to Jini.

NSF-JAPAN Award, 1999-2002. PI: Andre Scedrov.

DERA, 1998-1999. Instrumentation and Checking of Mobile Code.

ONR MURI Award, 1997-00 and continuation 2000-02. Semantic Consistency in Information Exchange; J. Mitchell, Principal Investigator. Additional participants: S. Kannan, I. Lee and A. Scedrov (Univ. Pennsylvania), R. Rubinfeld (Cornell), P. Lincoln (SRI), C. Dwork (IBM).

NSF Grant CCR-9629754, 1996-99. Object Systems: Programming Languages and Software Security; J. Mitchell, Principal Investigator.

TRW Foundation, 1994. Grant to study programming languages and system design languages; J. Mitchell, Principal Investigator.

NSF Grant CCR-9303099, 1993-96. Programming Language Analysis and Design; J. Mitchell, Principal Investigator.

NSF Presidential Young Investigator Award, 1988-93. Industrial funding provided by AT&T, Digital Equipment Corporation, Mitsubishi Corporation, the Powell Foundation and Xerox Corporation.

DARPA/ISTO BAA no.89-08, 1989-90, and continuations. Project to develop Common Prototyping Language and design accompanying Common Prototyping System. Principal investigators: David Luckham (Stanford) and Frank Belz (TRW).

NSF grant CCR-8814921, 1989-91. Research in Programming Language Structures: Types and Concurrency, J. Mitchell and V. Pratt, Principal Investigators.

NSF-INRIA grant for US-France collaboration, 1989-91. Val Breazu-Tannen, Carl Gunter principal investigators. Principal INRIA contacts: Gérard Huet (Paris) and Gilles Kahn (Sophia-Antipolis).

CNR (Italy) grant for Stanford-Italy collaboration, 1989-91. J. Barwise, S. Feferman, J. Mitchell (Stanford), M. Dezani (Turin), G. Longo (Pisa) principal investigators.

Honors

Wallace F. and Lucille M. Davis Faculty Scholar (1989–91),
NSF Presidential Young Investigator Award (1988–93),
IBM Graduate Fellowship (1983–84),
NSF Graduate Fellowship (1980–83),
Graduation with Distinction (Stanford 1978),
Juliet Knopp Lockwood Honors Scholarship (Stanford 1977),
Invitation to H^* honors mathematics program (U.W. 1975).

Professional Activities

Editorial Boards:

ACM Trans. Programming Languages and Systems, 1993 – 1996,
Chicago J. Theoretical Computer Science, 1994 – 2000,
Electronic Notes in Theoretical Computer Science, 1995 – 2000,
Information and Computation, 1987 – 2000,
J. Assoc. Computing Machinery (JACM), 1995 – 2000,
J. Computer Security, 2000 – present,
J. Functional Programming, 1989 – 2000,
Mathematical Structures in Computer Science, 1989 – 2000,
SIAM J. Computing, 1998 – present,
Theory and Practice of Object Systems, 1994 – 2000.

I resigned from several editorial boards en mass in 2000.

Special Issue Editor:

Info. and Computation: 1990 IEEE Logic in Comp. Sci. Conference.

General Chair:

IEEE Symp. on Logic in Computer Science, 1998-2001.

Conference Organizing Committees:

IEEE Symp. on Logic in Computer Science, 1990 – present,
Category Theory and Computer Science (organizing and program committee), 1990 – 1996.
Third Symp. Theor. Aspects of Computer Software (Advisory Committee) 1997.

Conference Program Chair:

IEEE Symp. on Logic in Computer Science, 1990,

Second Symp. Theor. Aspects of Computer Software, Sendai, Japan, 1994 (co-chair),
Advances in Types Systems for Computing, Newton Institute, Cambridge, 1995,
ACM Symp. on Principles of Programming Languages, 2002.

Conference Program Committees:

ACM Conf. Functional Programming and Computer Architecture (FPCA), 1989 and 1995.
ACM Conf. on Object-Oriented Programming (OOPSLA), 1988, 1992 and 1995.
ACM Symp. Principles of Programming Languages (POPL), 1989, 1991, 2000, and 2002.
ACM Workshop on ML, 1992.
Category Theory and Computer Science (CTCS), 1991, 1993, 1995 and 1997.
Conference on Automated Verification (CAV), 2000.
European Symposium On Programming (ESOP), 2002, 2003.
European Symposium On Research In Computer Security (ESORICS), 2000.
Foundations of Object-Oriented Languages (FOOL), 1995, 1996 and 1997.
Fundamentals of Computation Theory (FCT), 1999.
IEEE Computer Security Foundations Workshop (CSFW), 1999, 2002.
IEEE Symp. on Foundations of Computer Science (FOCS), 1988 and 1991.
IEEE Symp. on Logic in Computer Science (LICS), 1986, 1988, 1990 and 1996.
IEEE Symp. on Security and Privacy, 1999.
IFIP International Conference on Theoretical Computer Science (TCS), 2002.
Mathematical Foundations of Programming Language Semantics (MFPS), 1991, 1993, 1995.
Techniques of Object-Oriented Languages and Systems (TOOLS), 1992 and 1993.
Typed Lambda Calculi and Applications (TLCA), 1993.
Workshop on Formal Methods and Computer Security (FMCS), 2000.

Workshop and Program Organization:

- Workshop on Programming Languages and Security, DEC Systems Research Center, Oct 30-31, 1997. With M. Abadi (DEC, principal organizer), B. Bershad (U. Washington), E. Felton (Princeton), L. Gong (JavaSoft).

- Working group on Software Engineering and Programming Languages, CRA Meeting on Strategic Directions in Computing, on the occasion of the 50th anniversary of the ACM. Boston, June 14–15, 1996.
  Report: Gunter, C., Mitchell, J.C. and Notkin, D., Strategic Directions in Software Engineering and Programming Languages, *ACM Computing Surveys,* Vol 28A, No 4, December 1996.

- Workshop on Software Engineering and Programming Languages, sponsored by ARO and NSF, Boston, June 12–13, 1996. Co-organizer and program co-chair (with D. Notkin).

- Co-Organizer and meeting chair, ARPA-NSF workshop on Foundational Studies for Software Engineering, Stanford, Sept 6–7, 1995.

- Co-organizer, with S. Abramsky (Imperial), G. Kahn (INRIA) and A. Pitts (Cambridge), "Programme on Semantics of Computation," Isaac Newton Institute of Mathematical Sciences, University of Cambridge, U.K., July–December, 1995. This six-month program on involved approximately 65 visitors from North America, Europe and Japan, as well as hundreds of short-term participants in 10 special conferences and workshops at the Institute.
- Co-organizer, with D. Liddle (Interval), Stanford Workshop on Software Engineering, Sept 12–13, 1993.
- A founding organizer of "North-American Jumelage," annual workshop on programming language theory and logic in computer science, 1990–1995. Hosted first meeting at Stanford, 1990.

Column: Sigact News Logic Column, 1991 – 1997.

Regular Published Reviews:
Mathematical Reviews, Amer. Math. Society, 1989 – 1993
Zentralblatt für Mathematik/Mathematical Abstracts, Springer-Verlag, 1990 – 1993.

Member: IFIP Working Group 1.7, *Foundations of Security Analysis and Design* (Invited Charter Member), Assoc. Computing Machinery, Assoc. Symbolic Logic, European Assoc. for Theoretical Computer Science. Former member of IFIP Working Group 2.8, *Functional Programming* (Invited Charter Member).

INVITED LECTURES

1. Usenix Security Symposium, San Francisco, August, 2002.
2. Rewriting Techniques and Applications (RTA), Copenhagen, July, 2002.
3. Mathematical Foundations of Programming Semantics (MFPS), New Orleans, March, 2002.
4. IEEE Symp. Logic in Computer Science (LICS), Boston, June 2001.
5. European Symposium on Programming (ESOP), Genoa, Italy, April 2001.
6. Association for Symbolic Logic (ASL) Annual Meeting, Philadelphia, March 2001.
7. ACM Symp. Principles of Programming Languages (POPL), London, U.K., January 2001.
8. Invited panelist, Workshop on Challenges for Theoretical Computer Science (Organizers: David Johnson, Christos Papadimitriou, Avi Wigderson, Mihalis Yannakakis), Portland, OR, May 20, 2000.
9. Computer Science Logic (CSL '98), Brno (Czech Republic), August 24-28, 1998.

10. Marktoberdorf Summer School, 1998. Five lectures on security, network protocols and formal methods.

11. Conference on Computer-Aided Verification (CAV '98), June 28 - July 2, 1998, Vancouver, British Columbia.

12. Seventh CSLI Workshop on Logic, Language and Computation, May 29 - 31, 1998, Stanford University.

13. Linear '98, CIRM Luminy, Marseille, April 6–9, 1998.

14. ACM Workshop on Functional and Object-oriented Programming, July 2-7, 1997, Jadwisin (Poland)

15. Panelist, Software Engineering and Programming Languages, *Sigsoft '96,* ACM Symposium on Foundations of Software Engineering, 1996.

16. Marktoberdorf Summer School, 1996. Four lectures on type systems and object-oriented programming.

17. Linear '96, Tokyo, Japan. March 28–April 2, 1996. Lecture series on Decision and Optimization Problems in Linear Logic, with P. Lincoln and A. Scedrov.

18. Workshops and conferences associated with Isaac Newton Institute special program on Semantics of Computation:

    (a) Themes in the Semantics of Computation (org. S. Abramsky), July 17–21, 1995.

    (b) Linear Logic and Applications (org. G. Bierman), Oct 16–18, 1995.

    (c) Higher-order Techniques in Operational Semantics (org. A. Gordon) Oct 28-30, 1995.

    (d) Games, Processes and Logic (org. S. Abramsky), Nov 6–10, 1995.

19. Fundamentals of Computation Theory (FCT'95) Dresden, Germany, August 22–25, 1995.

20. EATCS Summer School on Typed Lambda Calculus and Functional Programming, Udine, Italy, September 20–30, 1994.

21. Second Symp. Theoretical Aspects of Computer Software, Sendai, Japan, April 18–21, 1994.

22. Invited tutorial, Techniques for Object-Oriented Languages and Systems (TOOLS) Conference, Santa Barbara. August, 1992.

23. Second Montreal Workshop on Programming Language Theory, Montreal, Quebec. December, 1991.

24. Sixth Workshop on Mathematical Foundations of Programming Semantics, Kingston, Ontario. May, 1990.

25. Japan Society for Software Science and Technology Annual Workshop on Object-Oriented Programming, Hakone, Japan, March, 1990.

26. MSRI Workshop on Logic from Computer Science, Berkeley, Nov., 1989.

27. Summer Conference on Category Theory and Computer Science (third biennial conference), Manchester, U.K., Sept., 1989.

28. IBM Distinguished Lecture Series on Semantics, April, 1989.

29. Special session on the semantics of inheritance, Fifth Workshop on Mathematical Foundations of Programming Semantics, New Orleans, March, 1989.

30. Annual Meeting of the Assoc. for Symbolic Logic, Los Angeles CA, January, 1989.

31. *Logic Colloquium '88,* Int'l meeting of the Assoc. for Symbolic Logic, Padova Italy, August, 1988.

32. CMU Workshop on the Semantics of Lambda Calculus and Category Theory, April, 1988.

33. Institute on Logical Foundations of Functional Programming, Univ. Texas Year of Programming, June, 1987. Organizer: G. Huet.

34. Institute on Encapsulation, Modularization and Reusability, Univ. Texas Year of Programming, April, 1987. Organizer: D. Gries.

35. Mid-Atlantic Mathematical Logic Seminar, Philadelphia, PA, February, 1987.

Publications

Books

1. J.C. Mitchell, *Concepts in Programming Languages,* Cambridge University Press, 2002, in press.

   J.C. Mitchell, *Foundations for Programming Languages,* MIT Press, 1996, 846 pages.

2. C.A. Gunter and J.C. Mitchell (eds.), *Theoretical Aspects of Object-Oriented Programming,* MIT Press, 1994, 548 pages.

Invited contributions to books

1. N.A. Durgin and J.C. Mitchell, Analysis of Security Protocols. In *Calculational System Design,* ed. M. Broy and R. Steinbruggen, IOS Press, 1999, pages 369–395. (Based on presentations at NATO Advanced Study Institute on Mathematical Methods in Program Development, Marktoberdorf, Germany, 1998.)

2. Mitchell, J.C., Hoang, M.K. and Howard, B., Labeling Techniques for Typed Fixed-point Operators. In *Higher-order Operational Techniques in Semantics,* ed. A.D. Gordon and A.M. Pitts, Publication of the Newton Institute, Cambridge Univ. Press, 1998, pages 137–174.

3. Fisher, K. and Mitchell, J.C., On the relationship between classes, objects and data abstraction. In *Mathematical Methods in Program Development,* Springer-Verlag NATO ASI series F (Computer and System Sciences), vol. 158, 1997, pages 371–408. (Proc. NATO Advanced Study Institute on Mathematical Methods in Program Development, Marktoberdorf, Germany, July 30 – August 11, 1996.)

4. Lincoln, P.D., J.C. Mitchell and A. Scedrov, Stochastic interaction and linear logic. In *Advances in Linear Logic,* ed. J.-Y Girard, Y. Lafont and L. Regnier, London Mathematical Society Lecture Notes, Volume 222, Cambridge University Press, 1995, pages 147–166. (Refereed)

5. Mitchell, J.C., On the equivalence of data representations. In *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy,* ed. V. Lifschitz, Academic Press, 1991, pages 305-330.

6. Kanellakis, P.C., Mairson, H.G. and Mitchell, J.C., Unification and ML type reconstruction. In *Computational Logic: Essays in Honor of Alan Robinson,* ed. J.-L. Lassez and G.D. Plotkin, MIT Press, 1991, pages 444–478.

7. Mitchell, J.C., Type Systems for Programming Languages. In *Handbook of Theoretical Computer Science,* ed. J. van Leeuwen, North-Holland, 1990, pages 366–458.

8. In *Logical Foundations of Functional Programming,* ed. Gérard Huet, Addison-Wesley, 1990:

(a) Mitchell, J.C., Polymorphic type inference and containment, pages 153–194,

(b) Mitchell, J.C., A type inference approach to reduction properties and semantics of polymorphic expressions (summary), pages 195–212,

(c) Bruce, K.B.,Meyer, A.R. and Mitchell, J.C., The semantics of second-order lambda calculus, pages 213–272,

(d) Meyer, A.R., Mitchell, J.C., Moggi, E. and Statman, R., Empty types in polymorphic lambda calculus, pages 273–284.

Refereed Articles in Archival Journals

1. N. Li, W. Winsborough, J.C. Mitchell, Distributed Credential Chain Discovery in Trust Management, J. Computer Security (accepted for publication).

2. Shmatikov, V. and Mitchell, J.C., Finite-State Analysis of Two Contract Signing Protocols, Theoretical Computer Science (accepted for publication).

3. Freund, S.N. and Mitchell, J.C., A Type System for Object Initialization in the Java Bytecode Language, ACM Trans. Programming Languages and Systems (accepted for publication).

4. Harper, R. and Mitchell, J.C., Parametricity and variants of Girard's J operator, Information Processing Letters 70, 1999, pages 1–5.

5. Lincoln, P.D., Mitchell, J.C. and Scedrov, A., Optimization complexity of linear logic proof games, *Theoretical Computer Science,* Special Issues on Linear Logic (accepted for publication).

6. Fisher, K. and Mitchell, J.C., On the relationship between classes, objects and data abstraction, *Theory and Practice of Object Systems,* Volume 4, number 1, 1998, pages 3–25. Invited paper for Special Issue on Third Workshop on Foundations of Object-Oriented Languages (held 1996).

7. Lincoln, P.D., Mitchell, J.C. and Scedrov, A., Linear Logic Proof Games and Optimization, *Bulletin of Symbolic Logic,* 2,3 (1996) 322-338.

8. Fisher, K., and Mitchell, J.C., The Development of Type Systems for Object-Oriented Languages, *Theory and Practice of Object Systems* 1,3 (1996) 189–220.

9. Mitchell, J. and Viswanathan, R., Standard ML-NJ weak polymorphism and imperative constructs, *Information and Computation* 127, 2 (1996) 102-116. Invited for special issue from IEEE Symp. Logic in Computer Science, 1993.

10. Fisher, K., Honsell, F. and Mitchell, J.C., A lambda calculus of objects and method specialization, *Nordic J. Computing* (formerly *BIT*) 1, 1 (1994) 3–37. Invited paper.

11. Mitchell, J.C., On abstraction and the expressive power of programming languages. *Science of Computer Programming* 212 (1993) 141–163. Invited for special issue of papers from Symp. Theor. Aspects of Computer Software, Sendai, Japan, 1991.

12. Cardelli, L., Martini, S., Mitchell, J. and Scedrov, A., An extension of System F with subtyping, *Information and Computation* 109 (1994) 4–56. Invited for special issue of papers from Symp. Theor. Aspects of Computer Software, Sendai, Japan, 1991.

13. Jategaonkar, L. and Mitchell, J.C., Type inference with extended pattern matching and subtypes, *Fund. Informaticae* 19 (1993) 127–166. Invited for special issue on type systems, ed. J. Tiuryn.

14. Lincoln, P., Mitchell, J.C., Scedrov, A. and Shankar, N., Decision Problems for Propositional Linear Logic, *Ann. Pure and Applied Logic* 56 (1992) 239–311.

15. Harper, R. and Mitchell, J.C., The type structure of Standard ML, *ACM Trans. Programming Languages and Systems,* 15, 2 (1993) 211–252.

16. Mitchell, J.C., Type inference with simple subtypes, *J. Functional Programming* 1, 3 (1991) 245–286.

17. Cardelli, L. and Mitchell, J.C., Operations on records, *Mathematical Structures in Computer Science* 1 (1991) 3–48.

18. Mitchell, J.C. and Moggi, E., Kripke-style models for typed lambda calculus, *Ann. Pure and Applied Logic* 51 (1991) 99–124.

19. Bruce, K.B., Meyer, A.R. and Mitchell, J.C., The semantics of second-order lambda calculus, *Information and Computation* 85,1 (1990) 76–134.

20. Mitchell, J.C. and Plotkin, G.D., Abstract types have existential type, *ACM Trans. Programming Languages and Systems,* 10, 3 (1988) 470–502.

21. Mitchell, J.C., Polymorphic type inference and containment, *Information and Computation* 76, 2/3 (1988) 211–249.

22. Dwork, C., Kanellakis, P.C. and Mitchell, J.C., On the sequential nature of unification, *J. Logic Programming* 1 (1984) 35–50.

23. Mitchell, J.C., The implication problem for functional and inclusion dependencies, *Information and Control* 56 (1983) 154–173. Abstract in *Zentralblatt für Mathematik* 539.68090 (1985).

24. Meyer, A.R. and Mitchell, J.C., Termination assertions for recursive programs: completeness and axiomatic definability, *Information and Control* 56 (1983) 112–138. Summary in *Zentralblatt für Mathematik* 537.68034 (1985).

25. Mitchell, J.C., Theilacker, J.C. and Klein, S.A., Calculation of monthly average collector operating time and parasitic energy requirements. *Solar Energy Journal* 26, 6 (1981).

Conference Publications (competitive selection based on 10 page technical summary)

1. N. Li, J.C. Mitchell, W.H. Winsborough, Design of a Role-based Trust-management Framework, IEEE Symp. on Security and Privacy, Oakland, May ???, 2002, to appear.

2. A. Chander, D. Dean, J.C. Mitchell, Deconstructing Trust Management, ACM SIGPLAN and IFIP WG 1.7 Workshop on Issues in the Theory of Security (WITS'02) Portland, Oregon, USA, January 14-15, 2002. Electronic proceedings at `http://www.dsi.unive.it/IFIPWG1_7/WITS200`

3. N. Li, W.H. Winsborough, J.C. Mitchell, Distributed Credential Chain Discovery in Trust Management, 8th ACM Computer and Communications Security Conference (CCS 2001), Philadelphia, Nov, 2001.

4. Comon, H., Cortier, V, and Mitchell, J.C., Tree Automata with one Memory, Set Constraints and Ping-Pong Protocols, ICALP 2001, Crete, Greece, July 8-12, 2001.

5. Chander, A., Mitchell, J.C., and Shin, I., Mobile code security by Java bytecode instrumentation, DARPA Information Survivability Conference and Exposition (DISCEX II), June, 2001.

6. Durgin, N.A., Mitchell, J.C., and Pavlovic, D., A Compositional Logic for Protocol Correctness, 14th IEEE Computer Security Foundations Workshop, Cape Breton, Nova Scotia, June 11-13, 2001.

7. Chander, A.. Dean, D, and Mitchell, J.C., A state-transition model of trust management and access control, 14th IEEE Computer Security Foundations Workshop, Cape Breton, Nova Scotia, June 11-13, 2001.

8. J. Mitchell, A. Ramanathan, A. Scedrov, and V. Teague, A probabilistic polynomial-time calculus for analysis of cryptographic protocols (Preliminary report), 17-th Annual Conference on the Mathematical Foundations of Programming Semantics, Arhus, Denmark, May, 2001, Electronic Notes in Theoretical Computer Science, Volume 45 (2001).

9. D. Lie, C. Thekkath, P. Lincoln, M. Mitchell, D. Boneh, J. Mitchell, M. Horowitz, Architectural Support for Copy and Tamper Resistant Software, Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX), Cambridge, MA, November 12-15, 2000.

10. I. Cervesato, N. Durgin, J. Mitchell, P. Lincoln and A. Scedrov, Relating Strands and Multiset Rewriting for Security Protocol Analysis, 13-th IEEE Computer Security Foundations Workshop, Cambridge, U.K., July 3-5, 2000.

11. Shmatikov, V. and Mitchell, J.C., Analysis of Abuse-Free Contract Signing, Financial Cryptography 00, accepted for publication.

12. Shmatikov, V. and Mitchell, J.C., Analysis of a Fair Exchange Protocol, Seventh Annual Symposium on Network and Distributed System Security (NDSS2000), accepted for publication.

13. Freund, S.N., and Mitchell, J.C., A Formal Framework for the Java Bytecode Language and Verifier, ACM Conference on Object-Oriented Programming: Systems, Languages and Applications, Denver, CO, November, 1999, pages 147-166.

14. Lincoln,P.D., Mitchell, J.C., Mitchell, M., and Scedrov, A., Probabilistic polynomial-time equivalence and security protocols, FM'99 World Congress On Formal Methods in the Development of Computing Systems, Toulouse, France, September, 1999.

15. Durgin, N.A., Lincoln, P.D., Mitchell, J.C., and Scedrov, A., Undecidability of bounded security protocols, Workshop on Formal Methods and Security Protocols (FMSP'99), Trento, Italy, July 5, 1999. Electronic proceedings: `http://www.cs.bell-labs.com/who/nch/fmsp99/prograı`

16. Cervesato,I., Durgin, N.A., Lincoln, P.D., Mitchell, J.C., and Scedrov, A., A meta-notation for protocol analysis, 12-th IEEE Computer Security Foundations Workshop, Mordano, Italy, June 28-30, 1999.

17. Bono, V., Patel, A., Shmatikov, V., and Mitchell, J.C., A core calculus of classes and mixins, European Conference on Object-Oriented Programming, 1999 (accepted for publication).

18. Bono, V., Patel, A., Shmatikov, V., and Mitchell, J.C., A core language of classes and objects, 15th Conf. Mathematical Foundations of Programming Semantics, 1999 (accepted for publication).

19. Mitchell, J.C., Mitchell, M., and Scedrov, A., A linguistic characterization of bounded oracle computation and probabilistic polynomial time, IEEE Foundations of Computer Science, Palo Alto, Ca. November 8–11, 1998, pages 725-733.

20. Lincoln, P.D., Mitchell, J.C., Mitchell, M., and Scedrov, A., A probabilistic poly-time framework for protocol analysis, 5th ACM Conference on Computer and Communications Security, San Francisco, Ca., November 3–5, 1998, pages 112–121.

21. Freund, S., and Mitchell, J.C., A type system for object initialization in the Java bytecode language, ACM Symp. Object-Oriented Programming: Systems, Languages and Applications (OOPSLA), Vancouver, October 20–22, 1998, pages 310-328.

22. Mitchell, J.C., Shmatikov, V., and Stern, U., Finite-State Analysis of SSL 3.0, Seventh USENIX Security Symposium, San Antonio, 1998, pages 201-216. Preliminary version presented at DIMACS Workshop on Design and Formal Verification of Security Protocols, September 1997, and distributed on workshop CD.

23. Agesen, O., Freund, S., and Mitchell, J.C., Adding Parameterized Types to Java, ACM Symp. Object-Oriented Programming: Systems, Languages and Applications (OOPSLA), Atlanta, October 7–9, 1997, pages 49-65.

24. Mitchell, J.C., Mitchell, M. and Stern, U., Automated Analysis of Cryptographic Protocols Using Murϕ, IEEE Symp. on Security and Privacy, Oakland, May 4–7, 1997, pages 141–151.

25. Lincoln, P.D., Mitchell, J.C. and Scedrov, A., The Complexity of Local Proof Search in Linear Logic (Extended Abstract). In *Proc. Linear Logic '96, Tokyo Meeting,* March 28–April 2, Tokyo, Electronic Notes in Theoretical Computer Science, Volume 3, 1996, 10 pages. `http://www1.elsevier.nl/mcs/tcs/pc/volume3.htm`. (Invited paper.)

26. Lincoln, P.D., Mitchell, J.C. and Scedrov, A., P. Lincoln, J. Mitchell, A. Scedrov "The Complexity Of Local Proof Search In Linear Logic" Workshop on Proof Search in Type Theoretic Languages, in association with CADE 13, New Brunswick, NJ, July 30 - August 3, 1996, pages 69–76. Proc. ed. D. Galmiche.

27. Mitchell, J.C. and Viswanathan, R., Effective models of polymorphism, subtyping and recursion, Proc. 23rd International Colloquium on Automata, Languages, and Programming (ICALP '96), Paderborn, Germany, July 8–12, Springer LNCS 1099, 1996, pages 170–181.

28. Fisher, K. and Mitchell, J.C., A delegation-based object calculus with subtyping, Proc. 10th Int'l Conf. Fundamentals of Computation Theory (FCT'95), Dresden, Germany, August 22–25, Springer LNCS 965, 1995, pages 42–61. (Invited paper.)

29. Hoang, M. and Mitchell, J.C., Lower bounds on type inference with subtypes, Proc. 22nd ACM Symp. on Principles of Programming Languages, San Francisco, CA, January 22–25, 1995, pages 176–185.

30. Fisher, K. and Mitchell, J.C., Notes on typed object-oriented programming, Proc. Theor. Aspects of Computer Software, Sendai, Japan, April 19–22, Springer LNCS 789, 1994, pages 844–885. (Invited paper.)

31. Katiyar, D., Luckham, D., and Mitchell, J.C., A type system for prototyping languages, Proc. 21st ACM Symp. on Principles of Programming Languages, Portland, January 17–21, 1994, pages 138–150.

32. Katiyar, D., Luckham, D., Meldal, S. and Mitchell, J.C., Polymorphism and subtyping in interfaces, ACM Workshop on Interface Definition Languages, 1994. Workshop associated with 21st ACM Symp. on Principles of Programming Languages, Portland, Oregon. Proceedings ed. J. Wing.

33. Mitchell, J.C., Honsell, F. and Fisher, K., A lambda calculus of objects and method specialization, Proc. 8th IEEE Symp. Logic in Computer Science, Montreal, June 19–23, 1993, pages 26–38.

34. Hoang, M., Mitchell, J.C. and Viswanathan, R., Standard ML weak polymorphism and imperative constructs, Proc. 8th IEEE Symp. Logic in Computer Science, Montreal, June 19–23, 1993, pages 15–25.

35. Mitchell, J.C. and Scedrov, A., Notes on sconing and relators, *Computer Science Logic '92, Selected Papers,* E. Börger et al., eds., Springer LNCS 702, 1993, pages 352–378. (Paper fully refereed after conference.)

36. Lincoln, P.D. and Mitchell, J.C., Operational aspects of linear lambda calculus, Proc. 7th IEEE Symp. Logic in Computer Science, Santa Cruz, June 22–25, 1992, pages 235–247.

37. Lincoln, P.D. and Mitchell, J.C., Algorithmic aspects of type inference with subtypes, Proc. 19th ACM Principles of Programming Languages Conf., Albuquerque, January 19–22, 1992, pages 293–304.

38. Bruce, K. and Mitchell, J.C., PER models of subtyping, recursive types and higher-order polymorphism, Proc. 19th ACM Principles of Programming Languages Conf., Albuquerque, January 19–22, 1992, pages 316–327.

39. Kurtz, S.A., Mitchell, J.C. and O'Donnell, M.J., Connecting formal semantics to constructive intuitions, in Myers and O'Donnell, eds., Constructivity in Computer Science Summer Symposium, San Antonio, June 19-21, Springer LNCS 613, 1992, pages 1–21.

40. Mitchell, J.C., On abstraction and the expressive power of programming languages, Proc. Theor. Aspects of Computer Software, Sendai, Japan, September 24–27, Springer LNCS 526, 1991, pages 290–310.

41. Cardelli, L., Martini, S., Mitchell, J. and Scedrov, A., An extension of System F with subtyping, Proc. Theor. Aspects of Computer Software, Sendai, Japan, September 24–27, Springer LNCS 526, 1991 pages 750–770.

42. Mitchell, J.C., Meldal, S. and Madhav, N., An extension of Standard ML modules with subtyping and inheritance, Proc. 18th ACM Principles of Programming Languages Conf., Orlando, January 21–23, 1991, pages 270–278.

43. Lincoln, P., Mitchell, J.C., Scedrov, A. and Shankar, N., Decision Problems for Propositional Linear Logic, Proc. 31st IEEE Symp. on Foundations of Computer Science, St. Louis, October 22–24, 1990, pages 662–671.

44. Howard, B. and Mitchell, J.C., Operational and axiomatic semantics of PCF, Proc. ACM Conf. Lisp and Functional Programming, Nice, France, June 27–29, 1990, pages 298–306.

45. Mitchell, J.C., Toward a typed foundation for method specialization and inheritance, Proc. 17th ACM Principles of Programming Languages Conf., San Francisco, January 17–19, 1990, pages 109–124.

46. Harper, R., Mitchell, J.C. and Moggi, E., Higher-order modules and the phase distinction, Proc. 17th ACM Principles of Programming Languages Conf., San Francisco, January 17–19, 1990, pages 341–354.

47. Canning, Cook, Hill, Mitchell and Olthoff, F-Bounded quantification for object-oriented programming, Proc. ACM Conf. Functional Programming and Computer Architecture, London, September 11–13, 1989, pages 273–280.

48. Cardelli, L. and Mitchell, J.C., Operations on records (summary), *Mathematical Foundations of Programming Language Semantics. Proceedings, 1989.* Springer-Verlag LNCS 442, 1990, pages 22–52. (Paper fully refereed after conference.)

49. Kanellakis, P.C. and Mitchell, J.C., Polymorphic unification and ML typing, Proc. 16th ACM Principles of Programming Languages Conf., Austin, January 11–13, 1989, pages 105–115.

50. Jategaonkar, L. and Mitchell, J.C., ML with extended pattern matching and subtypes, ACM Conf. on Lisp and Functional Programming, Snowbird, July 25–27, 1988, pages 198–211.

51. Mitchell, J.C. and Harper, R., The Essence of ML, Proc. 15th ACM Principles of Programming Languages Conf., San Diego, January 13–15, 1988, pages 28–46.

52. Mitchell, J.C. and Scott, P.J., Typed lambda models and cartesian closed categories, *Contemporary Mathematics,* Volume 92, Categories in Computer Science and Logic (Proceedings of a Summer Research Conference held June 14–20, 1987), Amer. Math. Society, 1989, pages 301–316. (Paper fully refereed after conference.)

53. Mitchell, J.C. and Moggi, E., Kripke-style models for typed lambda calculus, Proc. IEEE Symp. on Logic in Computer Science, Ithaca, June 22–25, 1987, pages 303–314.

54. Meyer, A.R., Mitchell, J.C., Moggi, E. and Statman, R., Empty types in polymorphic lambda calculus, Proc. 14th ACM Principles of Programming Languages Conf., Munich, January 21–23, 1987, pages 253–262.

55. Mitchell, J.C., A type inference approach to reduction properties and semantics of polymorphic expressions, Proc. ACM Lisp and Functional Programming Conf., Cambridge, MA, August 4–6, 1986, pages 308–319.

56. Mitchell, J.C. and O'Donnell, M.J., Realizability semantics for error-tolerant logics, Proc. Conf. on Theoretical Aspects of Reasoning About Knowledge, Monterey, CA, March 19–22, 1986, pages 363–382.

57. Mitchell, J.C., Representation independence and data abstraction, Proc. 13th ACM Principles of Programming Languages Conf., St. Petersburg, FL, January 13–15, 1986, pages 263–276.

58. Mitchell, J.C. and Meyer, A.R., Second-order logical relations, Proc. 1985 Logics of Programs, Brooklyn, June 17–19, Springer LNCS 193, 1985, pages 225–237.

59. Mitchell, J.C. and Plotkin, G.D., Abstract types have existential type, Proc. 12th ACM Principles of Programming Languages Conf., New Orleans, January 14–16, 1985, pages 37–51.

60. Mitchell, J.C., Semantic models of second-order lambda calculus, Proc. 25th Annual IEEE Symp. on Foundations of Computer Science, Singer Island, FL, October 24–26, 1984, pages 289–299.

61. Mitchell, J.C, Type inference and type containment. Proc. Int'l Symp. on the Semantics of Data Types, Sophia-Antipolis (France), June 27–29, Springer LNCS 173, 1984, pages 257–278.

62. Mitchell, J.C., Coercion and type inference. Proc. 11th ACM Principles of Programming Languages Conf., Salt Lake City, January 15–18, 1984, pages 175–185.

63. Mitchell, J.C., Inference rules for functional and inclusion dependencies. Proc. Second ACM Symp. on Principles of Database Systems, Atlanta, March 21–23, 1983, pages 58–69.

64. Meyer, A.R. and Mitchell, J.C., Axiomatic definability and completeness for recursive programs, Proc. 9th ACM Principles of Programming Languages Conf., Albuquerque, January 25–27, 1982, pages 337–346.

65. Mitchell, J.C., FCHART 4.0: The University of Wisconsin Solar Energy Design Program. Proc. DOE Systems Simulation and Economic Analysis Conference, January, 1980.

Workshop and Journal Abstracts

1. Freund, S. and Mitchell, J.C., A Type System for Object Initialization in the Java Bytecode Language. Overview of work in progress presented at Second Workshop on Higher-Order Operational Techniques in Semantics, December, 1997. Short summary appears in *Electronic Notes in Theoretical Computer Science* 19 (1998), `http://www.elsevier.nl/locate/entcs/volume10.html` 4 pages.

2. Mitchell, J.C., Typed lambda calculus and logical relations (abstract), *Journal of Symbolic Logic* (accepted for publication).

3. Mitchell, J.C., Abstract realizability for intuitionistic and relevant implication (abstract), *Journal of Symbolic Logic* 51, 3 (1986), pages 851–852.

Position papers

1. Gunter, C., Mitchell, J.C. and Notkin, D., Strategic Directions in Software Engineering and Programming Languages, *ACM Computing Surveys,* Vol 28A, No 4, December 1996.

2. Harper, R. and Mitchell, J.C., ML and Beyond, *ACM SIGPLAN Notices,* Vol 32, No 1, 1997, pages 80–85. Position statement on strategic directions for research in programming languages, in connection with Strategic Directions in Computing Research report on Programming Languages.

CONFERENCE LECTURES NOT ASSOCIATED WITH PUBLICATION

J. Mitchell, V. Shmatikov, U. Stern, Finite-State Analysis of SSL 3.0 and Related Protocols, DIMACS Workshop on Design and Formal Verification of Security Protocols, New Brunswick, September 3-5, 1997. (Informal publication on web site and CD-ROM.)

Predicative and Impredicative Polymorphism, Spring meeting of the Assoc. for Symbolic Logic, New York City, May, 1987.

Abstract realizability for intuitionistic and relevance logics, Assoc. Symbolic Logic Conf. on Logic, Language and Computation, Stanford CA, July, 1985.

Seminars at Universities and Research Organizations

AT&T Bell Labs
Brown University
Cambridge University
Carnegie-Mellon University
Columbia University
Cornell University
CUNY Graduate Center
Edinburgh University
IBM San Jose
IBM Yorktown Heights
Int'l Computer Science Inst. (Berkeley)
INRIA Rocquencourt (Paris)
Keio University (Tokyo)
MIT
Oxford University
Stanford University
SUNY Stony Brook
Toshiba Corporation (Kawasaki)
University of California, Berkeley
University of Chicago
University of Pennsylvania
Universita di Torino (Italy)
University of Wisconsin
Xerox PARC

Refereed Tutorials

Semantic Methods for Object-Oriented Languages, with Luca Cardelli, 1988 ACM Conference on Object-Oriented Programming: Systems, Languages and Architectures (OOPSLA), 1/2 day, 210 registrants.

PhD Students

Patrick D. Lincoln: Computational Aspects of Linear Logic. August, 1992. Present position: Director, Computer Science Laboratory, SRI International.

Brian T. Howard: Fixed Points and Extensionality in Typed Functional Languages. August, 1992. Present position: Assistant Professor, Bridgewater College.

Dinesh Katiyar: Theory and Practice of Typed Object-Oriented Programming. December, 1994. Subsequent positions: Software Engineer, Sun Microsystems; Founder & CEO, iLeverage (acquired by Epiphany).

Ramesh Viswanathan: Recursion Theoretic Semantics, Fully Abstract Term Models, and Imperative Constructs, June, 1995. Present position: Member Technical Staff, Lucent Bell Laboratories.

My Hoang: Type Inference and Program Evaluation in the Presence of Subtyping, June, 1995. Present position: Software Engineer, SAP International.

Ole Agesen: Type inference: A path to delivery of dynamically-typed object-oriented applications, December, 1995. (Co-advisee with David Ungar, Sun Microsystems.) Present position: Member Technical Staff, Sun Microsystems.

Kathleen Fisher: Type Systems for Object-Oriented Programming, August, 1996. Present position: Member Technical Staff, AT&T Laboratories.

Vitaly Shmatikov: Finite-State Analysis of Security Protocols, August, 2000. Present position: Member of Research Staff, Lucent Bell Laboratories.

Stephen Freund: Type Systems for Object-Oriented Intermediate Languages, December, 2000. Present position: Member of Research Staff, Compaq Systems Research Center.

Amit Patel: Entered Stanford Fall 1994. On leave to Google, Inc.; PhD expected 2001.

Mark Mitchell: Entered Stanford Fall 1996. On leave.

Nancy Durgin: Entered Stanford Fall 1998. PhD expected 2001.

Ajay Chandar: Entered Stanford Fall 1999. PhD expected 2002.

Vanessa Teague: Entered Stanford Fall 2000. PhD expected 2004.

S.M. Thesis Student

Lalita Jategaonkar: Supervisor of S.M. thesis research at AT&T Bell Labs under the MIT VI-A Cooperative Program. Thesis completed at MIT, 1989. Title: *ML with Extended Pattern Matching and Subtypes.* MIT supervisor: Albert R. Meyer.

Classroom Teaching

Stanford University:

Sophomore Seminar: Computer Security and Privacy (CS 99J)
Discrete Mathematics and Logic (CS 103x)
Intro. to Automata and Complexity Theory (CS 154)
Logic and Automated Reasoning (CS 157)
Discrete Structures and Algorithms (CS 161)
Programming Languages (CS 242)
Intro. to Programming Language Theory (CS 258)
Advanced Programming Languages (CS 342)
Topics in Programming Language Theory (CS 358).

CS 154, 157, 161, 258 and 358 are considered "theory" courses;
CS 242 and 342 are considered "systems" courses.

New York University: Type Theory and Programming Languages (G22.3033.01), with David MacQueen.

MIT: Teaching Assistant to A. Meyer and J. Stoy, Summer 1982. Lectured on the formal definition of Ada and principles of language design for special summer course on semantics of programming languages.
Teaching Assistant to B. Liskov, Fall 1981. Graduate course 6.821, *Concepts in Modern Programming Languages*.